A Simulation Framework for AI-Driven, Software Defined Network-Powered, Self-Healing 5G Security with Slice Isolation

Ajit Pillai[1], Akshita Shetty[2], Chaitravi Reddy[3], Ritisa Behera[4*] and Sanjay Vidhani[5]

[1,2,3,4,5]Department of Information Technology, KJ Somaiya School of Engineering, Mumbai, India

[1]ajit.pillai@somaiya.edu

[2]akshita.s@somaiya.edu

[3]chaitravi.r@somaiya.edu

[4]*ritisa.b@somaiya.edu*

[5]sanjayvidhani@somaiya.edu

Abstract. With the emergence of 5G networks, network slicing provides a virtualized network on shared infrastructure, increasing the attack surface to various threats, including DDoS attacks. Traditional security systems cannot handle the dynamic characteristics of 5G networks and therefore require a smart, automated solution. This paper introduces a simulation framework called Sentinel AI that combines Mininet, Ryu-based SDN control, and an optimized Random Forest classifier to enable real-time DDoS detection and automated self-healing mitigation in 5G networks using network slicing, without human intervention. The proposed system achieved 98.4% accuracy, 96.9% precision, 97.8% recall, a 0.7% false positive rate, and a 0.991 AUC-ROC for real-time DDoS detection across 6 test cases. Additionally, it achieved a 1.5-second average response time, more than 99.5% reduction in attack traffic, and 0% verified impact on other slices for automated mitigation across 6 different test cases. The proposed system operates at 60-70% CPU and 2.6 GB of RAM, even under high-attack conditions involving multiple slices, making it suitable for 5G network security research.

Keywords: 5G Network Slicing, Software Defined Networking, DDoS Detection, Machine Learning Security, Network Simulation, Ryu Controller, Random Forest Classification, Self-Healing Networks, Real-time Mitigation

B. Singh et al. (eds.), *Proceedings of the International Conference on Advances in Computing Technology and Artificial Intelligence (COMPUTATIA 2026)*, Atlantis Highlights in Intelligent Systems 18,
https://doi.org/10.2991/978-94-6239-713-2_33

1 Introduction

1.1 Background and Motivation

The advent of 5G mobile networks has revolutionized telecommunications infrastructure by offering data rates, ultra-low latency, and device connectivity previously unavailable [1]. Network slicing can be described as the core of the future 5G network paradigm, and the main purpose of network slicing is to allow the usage of multiple distinct networks that have certain characteristics along with a set of network requirements in a common network infrastructure. With the usage of network slicing, it is possible to allow the usage of multiple different network services, such as improved Mobile Broadband (eMBB) for broadband communication, Ultra Reliable Low Latency Communications (URLLC) for business-critical communication, and massive Machine-Type Communications (mMTC) for the Internet of Things [2].

The features of programmability and flexibility are achieved through a combination of capabilities from Software-Defined Networking (SDN) and Network Functions Virtualization (NFV). SDN architecture is unique in that it separates the control plane from the data plane. It can implement logical control centralization within a network and real-time network configuration using software-defined controllers [3]. The newly introduced SDN architecture enables a network operator to create, manage, and delete slices on demand. However, the ever-changing nature of 5G networks, as well as the shared infrastructure concept, also pose significant security challenges for 5G network slices and the entire 5G network [4]. The attack risk in 5G slices is much higher due to the coexistence of multiple logical networks that share the same physical resources. A security attack in one slice can compromise the integrity, availability, and confidentiality of other slices that share resources with it. Of all the different types of attack risks in the 5G slices, Distributed Denial of Service (DDoS) attacks pose one of the biggest risks to the 5G slices as a whole [5][1]. A DDoS attack can exhaust the resources of 5G slices, causing service outages across the entire 5G network.

These include traditional security measures such as static firewalls, intrusion detection systems, and manual response procedures, which are not very effective at addressing the challenges of 5G networks [4]. The high-speed and dynamic characteristics of 5G networks make it essential to have a real-time threat detection and mitigation facility that cannot be achieved through traditional security measures. Also, the resource boundaries between slices make it essential for security measures to effectively distinguish valid variations from malicious activities.

In particular, ML has revolutionized several research areas, such as computer vision, natural language processing, and network security, in which machines autonomously learn complex patterns from large datasets without explicit programming rules [6]. For instance, in network security, recent studies have shown that DDoS detection accuracy can be improved using ML-based methods, with deep learning achieving high detection rates in 5G networks, and that scalability and autonomy were identified as performance drivers for slice security frameworks [5][7]. The integration of flow-based feature sets, such as packet rate, protocol distribution, and inter-arrival time variance, into a hybrid classifier improves detection accuracy and response time by enabling the classifier to

leverage patterns across complementary feature dimensions [6]. When used with SDN-based network management paradigms, AI-based network security solutions can launch automated defense strategies to detect and respond to network threats by dynamically updating network policies, thereby avoiding the misclassification of legitimate network traffic [7][8].

1.2 Problem Statement

Although considerable research has addressed 5G security, network slicing, and AI-driven threat detection, gaps remain in the development of pragmatic, integrated frameworks for 5G slice environments [9]. There are a number of different challenges that have led to this research. One such challenge is that, given the need to enforce isolation, runtime policies must adapt to changing network conditions while maintaining strict separation between slices to avoid cross-slice attacks [4]. Secondly, 5G networks have high throughput requirements; hence, there is a need for real-time threat detection, enabling systems to analyze packet flow with minimal latency and identify patterns before damage is done. Thirdly, given the high-speed requirements of 5G, it is untenable to rely on manual intervention and to require systems to automatically mitigate threats by intelligently modifying policies and blocking malicious sources without any human involvement [10], [8]. Moreover, systems must be self-healing so they can automatically counterattack and recover from attacks, and they must be scalable to handle high volumes of traffic without compromising performance. Situational awareness is also a key aspect, so enabling operators to monitor security activities and make decisions is important.

1.3 Research Objectives

This paper addresses the identified challenges in AI-driven security for SDN-based 5G slices by developing a comprehensive simulation framework called SentinelAI. The objectives of the research in this paper can be summarized as follows:

- **Framework Architecture:** Design a modular simulation framework that is scalable and integrates SDN control functionality, threat analysis using machine learning algorithms, and mitigation techniques for analysis of 5G slice security [11].
- **Traffic Simulation:** Realistic traffic-generation models will be designed to accurately represent the security characteristics of 5G slices, such as eMBB, URLLC, and mMTC traffic.
- **AI-driven Detection:** The implementation and evaluation of machine learning algorithms, particularly improved versions of Random Forest algorithms, for detecting DDoS attacks in a high-speed environment in real-time [6].
- **Automated Mitigation:** Design and implement a self-healing mitigation process that can automatically respond to detected threats using IP blocking, rate limiting, and flow rule changes to ensure continued service for legal traffic [4].

- **Real-time Monitoring:** Design a complete solution for monitoring and visualization that provides operators with real-time information on network activities and security events via an intuitive web-based interface.
- **Performance Evaluation:** Extensive simulation experiments are conducted to evaluate the efficiency of the designed framework across various parameters, such as mitigation time, detection rate, and false-positive rate relative to legal traffic.

1.4 Contributions

The main contributions of this study to the area of 5G network security are: the introduction of SentinelAI, a comprehensive end-to-end security solution that is fully focused on providing security for slice-based 5G networks and integrates traffic monitoring, AI-driven threat detection, and mitigation in a single solution, the design and implementation of a highly flexible simulation-based evaluation solution that utilizes industry-standard tools such as Mininet, Ryu, and Scapy for comprehensive testing and evaluation of security solutions in a controlled environment; the design and implementation of a highly improved Machine Learning (ML) solution that utilizes a highly optimized Random Forest classifier and utilizes traffic features that are engineered to provide highly accurate results in anomaly detection, suitable for the high-speed nature of 5G networks; the design and implementation of a practical solution for automated self-healing mechanisms, where dynamic flow rule generation and intelligent IP blocking are performed in a fully automated and human-intervention-free manner, and SentinelAI, where security policies are fully enforced and the isolation boundary is strictly maintained while providing a solution that is capable of implementing coordinated defense strategies that involve multiple slices [8].

1.5 Organization of the Paper

The rest of the paper is divided into the following sections. Section II of the paper focuses on related work, discussing 5G security, network slicing, SDN security management, and AI-based threat detection, all of which motivate this research. Section III of the paper focuses on the detailed architecture of the proposed SentinelAI framework, including the SDN framework, the traffic simulation environment, the machine learning process, and the mitigation techniques. Section IV of the paper focuses on the implementation of the proposed system, including the technology used, the implementation process, and the system's integration. Section V of the paper presents a detailed analysis of the experimental results, including the detection, mitigation, and efficiency of the proposed system across different attack types. Section VI of the paper focuses on the analysis of the results, the shortcomings of the proposed system, and the future scope of the research. Section VII presents the research conclusion, including an introduction to the proposed contributions and their significance for the evolution of 5G secure networks.

2 Literature review

2.1 5G Network Slicing and Security Challenges

Another major innovation in 5G technology is network slicing, which enables network operators to create customized logical networks over a common physical network infrastructure [12]. However, network slicing also raises numerous security concerns. Previous studies have focused on integrating Machine Learning (ML) with SDN and NFV to ensure security across network slices. For example, the authors [2] have shown that ML can be effectively combined with SDN and NFV to ensure security in network slices. However, they have also identified security concerns associated with real-time responses to security threats. In another study, the authors [7] conducted a systematic literature review to identify ML-based approaches to secure network slicing in 5G networks. However, they have identified security concerns that are associated with scalability and a lack of autonomy in network slices. Although it is a conceptual study, it lacks practical validity.

The authors [3] have emphasized that traditional security models are not appropriate for 5G networks due to dynamic trust relationships and the need for continuous monitoring of network slices.

2.2 SDN-Based Security Management

Software-Defined Networking (SDN) is identified as one of the most important technologies for ensuring dynamic and programmable security management for emerging network architectures. Conventional SDN-based security management relies on external security devices that monitor network traffic and interact with the SDN controller to enforce network security. This may, however, lead to performance issues, including latency, and is therefore not considered suitable for time-critical 5G services.

To overcome these limitations, recent research studies have proposed integrating security intelligence into SDN controllers. Several approaches have been proposed, including anomaly-based intrusion detection, supervised traffic classification, flow-based policy enforcement, and the deployment of honeypots and honeynets for intrusion detection and analysis [2] [7].

In addition, the authors have proposed several methods to ensure secure, resilient, and differentiated 5G network operations, including SDN-based enforcement of fine-grained network security policies based on traffic classification and service requirements [12].

The programmable nature of SDN is considered one of the most important advantages in ensuring robust and sophisticated defense strategies, such as deploying honeypots, modifying network topology, and allocating network resources to mitigate resource exhaustion attacks. Escolar, Wang, and Calero proposed a honeynet framework using SDN for large-scale IoT services in smart cities, ensuring robust security for emerging network architectures [18].

2.3 SDN-Based Defense and DDoS Mitigation

The role of SDN in facilitating effective dynamic defense mechanisms against Distributed Denial of Service (DDoS) attacks in 5G networks cannot be overstated. In general, security mechanisms in traditional systems often involve dedicated hardware solutions, which introduce delays in the network, a factor that cannot be allowed in 5G systems.

To overcome the limitations of introducing delays in the network, recent studies have proposed integrating security intelligence into the SDN control plane. The authors showed that SDN can be used to perform fine-grained traffic management and to coordinate distributed DDoS attack defense mechanisms without introducing significant delays in the network [12]. The programmability of the SDN data plane complements the system by enabling dynamic responses to DDoS attacks.

2.4 Machine Learning for Network Security

The increasing complexity of cyber threats has made traditional signature-based methods inadequate for detecting modern cyber threats; hence, the application of ML to improve network security. A detailed survey on ML-based solutions for 5G networks demonstrated the efficacy of supervised classification and anomaly-based detection methods.

The major trade-off in ML-based network security solutions, as reported in the literature, was between accuracy and latency. Although deep learning methods such as CNNs and RNNs achieve high accuracy in ML-based network security solutions, they are computationally intensive and introduce latency.

On the contrary, Random Forest-based ML solutions provide a good trade-off between accuracy and computation efficiency. They perform well for high-dimensional feature spaces and are efficient for real-time decision-making, making them a good candidate for ML-based solutions for 5G network security, as used in this work.

2.5 Simulation Frameworks for Network Security

Simulation environments play a critical role in evaluating various security mechanisms in ethical and practical contexts. Mininet is now an accepted tool for research in SDN environments owing to its ability to simulate real-world networks on a single machine.

While tools such as ns-3 and OPNET allow detailed modeling of various lower-layer protocols, they also introduce added complexity and reduced flexibility in SDN-based environments. Mininet is ideal for evaluating controller logic and flow-based environments, though it is more focused on functional accuracy than on precise timing.

At this moment, it is apparent that there is a lack of comprehensive frameworks that integrate various aspects of network slicing, SDN-based policy enforcement, and AI-based threat detection. As such, researchers have resorted to developing their own solutions.

2.6 Gap Analysis and Motivation

The review of existing literature has resulted in the identification of various gaps that are critical for the development of SentinelAI:

Integration Gap: The existing literature focuses on threat detection, mitigation, and policy enforcement methods in isolation. However, a unified framework for integrating all these components is still unavailable [8].

- **Real-Time Performance Gap:** Although ML-based methods achieve high accuracy in detecting various network threats, they do not meet the real-time performance demands of 5G technology.
- **Self-Healing Gap:** Most existing methods require human intervention for the final decision-making process. However, self-healing-based autonomous methods for network security are still in their initial stages.
- **Transparency Gap:** The existing literature regarding AI-based methods for network security considers them as "black boxes."

3 System Architecture

3.1 Overview

The overall principles of SDN are followed in the SentinelAI framework. Hence, the SentinelAI framework is designed as a three-tiered model comprising a Data Plane, a Control Plane, and an Application Plane [13]. Such a design makes the overall development process modular and enables scaling and the addition of new functionality without affecting existing functionality, as shown in Fig. 1.

The Data Plane consists of a set of virtual 5G infrastructure components, including a virtual switch, hosts for different slice types, and a traffic-generation component. Furthermore, the Control Plane consists of a Ryu SDN controller that manages the entire process. Finally, the Application Plane consists of a machine-learning-based threat engine and a web-based monitoring tool for visibility into overall system operations.

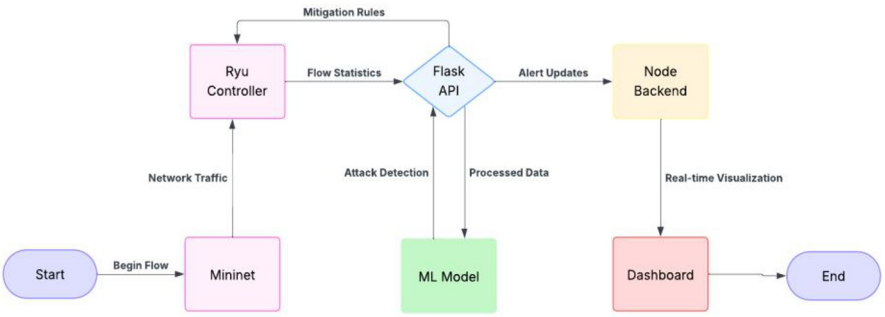

Fig. 1. Overall system architecture of SentinelAI, integrating traffic emulation, AI-driven detection, SDN mitigation, and real-time visualization.

Communication among these planes is facilitated by SDN protocols, where OpenFlow allows the controller to communicate with other devices in the network, and by communication between the controller, ML engine, and dashboard via RESTful APIs

and WebSocket connections. This is standardized to enable interoperability and facilitate further integration with other SDN-based systems [12].

3.2 Data Plane Architecture

The Data Plane consists of physical and logical network infrastructure that carries user traffic and enforces forwarding policies determined by the Control Plane. In the SentinelAI framework, the Data Plane is simulated with Mininet, which provides a realistic network environment with virtual switches and hosts [11]. The load tests are carried out by using Locust to mimic high-volume DDoS attacks and normal traffic patterns [5].

The network topology simulated above is a simple representation of a 5G core network architecture with multiple network slices. The network slices are logically isolated by implementing Virtual Local Area Network (VLAN) tagging and flow table rules that define isolation boundaries. The network topology consists of:

- An ingress component simulating the 5G User Equipment (UE) interface
- A primary switch for initial traffic classification and slice selection
- Secondary switches for different types of network slices: eMBB, URLLC, and mMTC
- Host components simulating servers and services for each network slice
- A traffic monitor tap that replicates traffic to the detection engine

The simulated components are equivalent to the 3GPP Release 16 equivalent entities in the traffic generator, with 50 gNB identifiers (gNB_001 to gNB_050), corresponding to the next-generation NodeB user plane interfaces; 5 UPF identifiers (UPF_01 to UPF_05), corresponding to the per-slice User Plane Function entities; 10 AMF identifiers (AMF_01 to AMF_10), corresponding to the Access and Mobility Management Function entities; VLAN tags corresponding to S-NSSAI identifiers (in the form of NSSAIs with SST and SD fields); and 5QI values (5QI 2, 3, 6, 7, 9), corresponding to quality of service Class Identifier assigned per slice type, with URLLC at 5QI 3 with 10 ms packet delay budget, eMBB at 5QI 6 with 300 ms, and mMTC at 5QI 9 with 1000 ms. The 5G core entities not considered in the above simulation are PFCP-based UPF control signaling, NAS-layer procedures, and inter-AMF handover signaling.

Realistic traffic scenarios that mirror real-world 5G environments are important for testing and validating the security mechanisms in place. To this end, the architecture utilizes specific traffic and attack simulators. Each slice type has its own flows, and legitimate traffic generators are used to create them. The eMBB traffic is simulated by generating variable-sized packets and bursty traffic, while the URLLC traffic is simulated by generating small packets at regular intervals, requiring strict timing requirements. In mMTC, a large number of small, infrequent packets are sent from multiple source addresses, simulating IoT devices [14].

Different DDoS attack patterns are simulated by the attack simulators, and most are User Datagram Protocol (UDP) flood attacks, given their popularity and effectiveness against network infrastructure. The attack simulator can alter packet rate, payload size, and source address patterns to simulate different attack strategies, such as single-source

floods, distributed attacks from multiple sources, and amplification attacks, using Locust for large-scale load generation [9].

Slice Isolation Mechanisms: Isolation between slices is necessary to prevent attacks in one slice from propagating to other slices. The data plane has different mechanisms for isolation.

- Logical isolation with VLANs assigns a different VLAN to each slice, hence providing Layer 2 isolation between the slices [1].
- Rules installed by the controller in the flow tables provide strict forwarding rules that do not allow cross-slice communications unless configured.
- Quality of Service (QoS) mechanisms are used to ensure that a given slice cannot consume all the available bandwidth, hence starving other slices [10].

3.3 Control Plane Architecture

The Control Plane is the brain of the network management system, as well as the orchestration point of the security policies. The Ryu SDN controller is used to implement the Control Plane functionality in SentinelAI, facilitating DDoS attack mitigation through OpenFlow rules, as mentioned in reference [3].

3.4 Ryu Controller Configuration

Ryu is a software controller that operates using Python and communicates with network switches via the OpenFlow protocol. This is a configuration of a controller, which has a number of different application modules that implement different aspects of network management:

- The topology discovery module is used to continually update its view of network topology by processing Link Layer Discovery Protocol packets and switch feature responses. This allows it to make intelligent forwarding decisions and identify potential attempts to manipulate the network topology, which could indicate a network attack [13].
- The flow management module is used to install, update, and delete flow entries in a flow table used by network switches. High-priority flow rules block network attacks and take precedence over normal forwarding operations, depending on the nature of the security policies [7].
- Slice Management Module is used to implement policies related to different types of network slices, including bandwidth, quality of service, and isolation. This module integrates with the ML detection engine and implements slice-based security policies that take into account the unique characteristics of each slice [2].

Policy enforcement: When the ML engine identifies malicious traffic, it notifies the controller of the threat details. This includes the source IP address, the type of attack, and the affected slice. The controller then uses various countermeasures:

- In cases where confirmed attacks occur, for instance, in DDoS attacks, high-priority flow rules are implemented. This action drops all packets from a particular source IP address used in the attack. Such blocking rules are implemented as early as possible in the network path in order to limit the resource consumption of malicious traffic [15].

- The implementation of rate-limiting policies can also occur in cases where a threat is suspected but not confirmed. In this instance, traffic from particular sources is limited to a particular level. This is done to reduce the false positive rate when the ML model is not very confident in its decision [16].
- Traffic redirection policies can redirect particular flows to honeypot systems for deeper analysis. Such an action is undertaken to gather threat intelligence on attack types while ensuring that specific communications are not disrupted [19].

Dynamic Rule Management: The controller maintains a database of all active security rules, with timestamps indicating when the rules were installed and statistical data on the number of packets a particular rule has matched. This allows for a number of important capabilities, such as:

- Automatic expiration of the rules, which prevents the blocking of addresses that may have been temporarily compromised. After a defined period, the blocking rules are automatically terminated, and traffic from previously blocked addresses is again analyzed to determine whether the threat has dissipated [4].
- Statistical analysis of the number of matches for each rule, which is used in the identification of patterns and effectiveness in the mitigation process. High matches for a particular rule indicate that the attack is still ongoing, whereas zero matches indicate a false positive, warranting further analysis [20].
- Rule optimization, where obsolete and unnecessary rules that consume valuable space in the flow tables are identified and eliminated, given that space is limited and that important security rules must always be accommodated in case a prolonged attack is experienced [18].

3.5 Application Plane Architecture

The Application Plane is a representation of the intelligent layer, which processes network behavior to identify potential threats and display information to network operators. The major components of SentinelAI include an ML-based detection engine and a web-based monitoring dashboard, with Node.js for API management and ML serving via Flask, a Python-based web framework [6].

Machine Learning Detection Engine: The engine utilizes an ensemble of machine learning algorithms, such as but not limited to, Random Forest, XGBoost (eXtreme Gradient Boosting), LightGBM (Light Gradient Boosting Machine), LSTM (Long Short-Term Memory), and SVM (Support Vector Machine) trained on labeled network traffic data to effectively implement the threat identification logic for DDoS classification. In the current implementation, a random forest classifier is being utilized for DDoS detection. This is because the random forest classifier can handle high-dimensional data and is fast enough for real-time applications. In addition, the random forest classifier is set to have 100 decision trees and a maximum depth of 20. Furthermore, to address the class imbalance in the dataset, class_weight is set to 'balanced' to ensure the minority classes are considered during model training. The model is pre-trained and loaded during runtime using Joblib. The overall architecture of the engine comprises a number of stages:

- The first step is packet capturing and feature extraction, wherein Scapy or Pyshark is used for packet capturing and feature extraction. Features include statistical measures over a specified period, packet count, byte count, packet rate, protocol information, and behavioral information, resulting in more than 17 flow-based attributes [21].
- Preprocessing of the features takes place for the purpose of ensuring consistency and reliability while making any prediction by the model. Initially, the raw features based on packet information are aggregated and converted into flow-based statistical features, which include the number of packets, the number of bytes, the packet rate, the average size of the packets, the protocol used for communication, the variance of interarrival time, and the entropy of ports used for communication. In total, 34 raw features are extracted; features with a Pearson correlation coefficient greater than 0.95 are eliminated to reduce redundancy, resulting in 31 features. These features are then normalized using StandardScaler to ensure consistency with the training model. Incomplete information regarding the packets is handled during the preprocessing phase. [17].
- The data set used for training and evaluation is artificially created and has 24,000 labeled samples of network traffic, consisting of both normal and malicious patterns of traffic. The dataset comprises 8 classes of traffic patterns: normal traffic and 7 attack types, namely TCP SYN Flood, HTTP Flood, UDP Amplification, ICMP Flood, Slowloris, DNS Tunneling, and QUIC Flood. Each class in this data set has 3,000 samples. The data set is imbalanced, with 7 times as much malicious data as normal data (87.5% malicious).
- Classification uses ensemble models on preprocessed features to classify network traffic as normal or a DDoS attack. The classification results are obtained by passing preprocessed features to a model, which returns a classification label and a confidence score. The latency achieved by models is below 50ms [9]. The training process uses an 80-20 stratified split of the data, with 19,200 samples for training and 4,800 for testing. The stratified split of the data ensures that each traffic class is well represented in both the training and test sets. During training, class imbalance is addressed using class-weighting techniques. After training, the model is serialized and integrated into the detection engine, where it is used for real-time traffic detection with a latency of under 50 milliseconds.
- Business logic rules, which are essentially a form of post-processing classification results, are used before initiating mitigation actions. This includes a threshold value that indicates the minimum probability required to ensure reliable classification results. If network traffic is classified as malicious, network traffic from previously whitelisted sources can be allowed to pass [4]. If ML models fail to perform well, rule-based systems can be a fallback.
- The performance of the detection model is measured using standard metrics for evaluating a classification model, such as accuracy, precision, recall, F1-score, and Area Under the Receiver Operating Characteristic Curve (AUC-ROC) for measuring detection and impact on normal traffic. The false positive rate (FPR) is also used to assess the likelihood of misclassifying normal traffic

as malicious. This ensures a comprehensive evaluation of the detection model's performance in real-world 5G network scenarios.

Dashboard and Visualization: The dashboard provides real-time visibility into network activity and security incidents through an intuitive interface, specifically crafted for security operations teams, built with React for the UI and Node.js for back-end service orchestration [10].

The main dashboard view presents a number of information boxes that are updated in real-time via WebSocket connections:

- On the other hand, a traffic visualization panel displays graphs of the volume of valid and malicious traffic over time, thus helping the network operator to determine when an attack has begun, its severity, and how well it has been mitigated.
- A table of packets displays the packets received recently, including their time of receipt, their source and destination, their protocol, their assigned slice, and their classification status. The malicious ones are color-coded to make them easy to identify.
- An alerts panel displays various network security alerts, including detected attacks, mitigation, and status change alerts. Each alert includes its time, severity, and description, thereby facilitating detailed analysis of the attack.
- A blocked IP management panel displays a list of IP addresses blocked, including reasons for the block and the time to expiration of the block. It is also from this panel that the network administrator can remove an IP address from the blocked list if analysis indicates it should not have been blocked.
- A slice status panel displays the status of network slices, including their current traffic, number of flows, threats, and mitigation status. This panel helps the network administrator determine whether the attack has targeted specific slices or the entire network [14].

3.6 Security Workflow

The Sentinel AI security workflow utilizes all three planes of the architecture to provide comprehensive security.

- **Normal Operation:** Data transfer occurs through the Data Plane along routes determined by flow rules configured by the Control Plane. The ML engine always checks the flow for feature extraction and classification. If it classifies the flow as normal, it does nothing but update statistics and dashboard displays [3].
- **Threat Detection:** If the ML engine classifies the flow as malicious with a high confidence level, it sends out an alert message to both the Control Plane and Application Plane with all information regarding the threat in terms of source and destination IP addresses, type of attack, slice affected, confidence level, and recommended mitigation [22].
- **Automated Mitigation:** Upon receiving the threat alert, the Control Plane checks its security policy to determine whether the recommended mitigation should be implemented. After approval, it adds high-priority flow rules to mitigate the threat. Address-based threat mitigation would involve a single rule at the switch to filter out the source IP and drop all packets to it. More complex

threat mitigation may require multiple rules across different switches to provide complete security [5].

- **Continuous Monitoring:** After the implementation of the mitigation rules, the detection engine continues to monitor traffic to assess the effectiveness of the mitigation. It also detects whether the attacks have started coming from different source addresses or attack vectors. The dashboard displays the status of the mitigation in progress, the number of packets blocked, and the reduction in attack traffic [9].
- **Recovery and Cleanup:** After the attack traffic reduces, either because the attack was successfully mitigated or the attackers have stopped their attack, the system initiates the recovery process. The controller will automatically delete the flow rules that were set with expiration timeouts. The ML engine continues to monitor whether the source addresses that previously exhibited malicious behavior have returned to normal or remain malicious. The dashboard also provides the operator with historical information about the attack incident to assist in analyzing post-incident activity [4].

4 Implementation

The Sentinel AI framework has been fully implemented as a complete system that can be deployed on any laptop or desktop computer. The framework has been implemented using only open-source tools to make it easily reproducible and easily extensible by other researchers. The implementation includes all components of network emulation, SDN control, machine-learning detection, mitigation, and visualization, in close integration with one another through well-defined interfaces [11].

The network infrastructure uses Mininet 2.3.0 with Ubuntu 20.04 LTS as the OS, running in Oracle VM VirtualBox 6.1. Mininet creates a realistic virtual network with a primary switch connected via OpenFlow to the Ryu controller, and three switches for each of the primary 5G service classes—eMBB, URLLC, and mMTC—each with a unique VLAN tag (VLAN 10, 20, and 30). Each slice has numerous hosts to simulate user equipment and application servers. The monitoring host connects to each slice via a mirror port, which forwards all packets passing through the network to the detection module for analysis without impacting normal operation. Open vSwitch 2.15 serves as the software switch due to its excellent OpenFlow 1.3 support and compatibility with Ryu [13].

The Ryu SDN Framework version 4.34, with Python 3.9, is used for the SDN network's control plane. Various modules are used for the controller application, including topology discovery and maintenance using LLDP packets, default flows for basic connectivity, ARP requests, strict slice isolation by dropping all inter-slice traffic, quality of service and bandwidth policing for different types of slices, RESTful API server on port 8080 for receiving alerts from the ML engine and status requests for the dashboard, and dynamic mitigation logic for installing high-priority drop or metering rules on receiving the threat notifications [5]. Hard timeouts are applied to all rules so the network can heal itself after the attack, with the option to manually unblock via the dashboard [4].

The ML detection pipeline is a separate Python process that always receives mirrored traffic using Scapy 2.4.5. The pre-trained model, a Random Forest model with 100 trees and a maximum depth of 20, is loaded during startup using Joblib. The model is trained on a synthetic dataset comprising 24,000 samples across 8 traffic classes, with 3,000 samples per class. There is an 80/20 stratified split between the training and test data sets, comprising 19,200 and 4,800 samples, respectively. The proportion of malicious to normal is 7:1, or 87.5% of the data set is malicious, achieved by using class_weight='balanced' for the Random Forest model and scale_pos_weight for the XGBoost model. Of the 34 raw features, those with a Pearson correlation > 0.95 were recursively removed, resulting in 31 features used for classification, as listed in 5g_feature_names.pkl. Every 2 seconds, all captured packets are processed to generate a feature vector containing more than 20 statistical features, such as packet counts, packet sizes, packet rates, average packet sizes, protocol types (TCP/UDP/ICMP), small packet ratios, inter-arrival time variance, flow durations, source/destination port entropy, and flag types. The classifier then uses the same StandardScaler technique applied during training to generate a probability score. When the score is above 0.85, an HTTP POST request containing the source IP address of the malicious packet, affected slice, confidence level, and recommended action is sent to the Ryu controller's REST API. However, the entire process is designed to ensure reliability in the event of packet loss or controller unavailability, with retries logging details instead of terminating [17]. The real-time monitoring and operation dashboard is a full-stack web application that uses Node.js 16 as the back-end runtime, Express.js as the back-end framework, and React 18.2 (Vite-built) as the front-end framework. The back-end utilizes WebSocket connections through Socket.io 4.5 to send real-time updates whenever the controller installs or uninstalls a flow rule, the ML engine sends an alert, or the application receives a manual unblock request [10]. The real-time dashboard has five main components:

1. An interactive Chart.js line graph displaying the volume of normal traffic as well as malicious traffic over time, segmented by slices.
2. A scrolling table displaying the most recent 10,000 packets, colored according to the traffic type (green for normal, red for malicious).
3. A chronological feed of severity-badged alerts.
4. A table displaying blocked IP information, including the unblocking reason, time, expiration, and a one-click unblock option.
5. A visual representation of the health of each slice in terms of current throughput, active flows, and active mitigations.

The entire dashboard is responsive, and the theme is dark, which is suitable for 24/7 operation centers of security companies. Integration testing has been performed to verify the smooth interaction of all components of the system. A Bash orchestration script has been designed to launch Mininet with the correct topology, the Ryu controller with all required apps, the Scapy-based detection engine, and finally the Node.js dashboard server. The health-check endpoints and restart features ensure that, in the event of a crash of any component, the entire system will not crash [14].

5 Experimental Results

To test and determine the effectiveness of SentinelAI, six unique test scenarios have been created, as shown in Table 1, ranging from a clean baseline with no illegitimate traffic to sophisticated adaptive and application-level attacks.

Table 1. Test Scenarios

Scenario	Description
Baseline	Only legitimate traffic, all slices
Single-slice attack	UDP flood targeting one slice
Multi-slice attack	Simultaneous attacks on 2–3 slices
Adaptive attack	Gradually increasing rate + source IP rotation
TCP SYN flood attack	Targeting the eMBB slice, exploiting TCP handshake exhaustion at 5,000–15,000 pps with packet sizes of 60–100 bytes
Application-layer attack	Targeting the URLLC slice at low packet rates (1–10 pps for Slowloris, 800–1200-byte payloads for HTTP flood)

The performance of SentinelAI's detection capabilities across all six scenarios was measured using standard classification metrics. As shown in Table 2, the system's overall accuracy is 98.4%, precision is 96.9%, and overall recall is 97.8%. The false positive rate is as low as 0.7%. The overall F1-score is 0.973, and the AUC-ROC is 0.991. These figures indicate that, overall, there is no significant disruption to normal traffic flow, and SentinelAI's detection capabilities remain high. The top-3 most discriminative features identified by the classifier are packet rate, protocol distribution, and average packet size, as reported by [6] for 5G threat detection using statistical features at the flow level.

Table 2. Detection Performance

Metric	Value
Accuracy	98.4%
Precision	96.9%
Recall (TPR)	97.8%
False Positive Rate	0.7%
F1-Score	0.973
AUC-ROC	0.991

Top-3 features: packet rate → protocol distribution → average packet size.

The framework was tested against 8 different attack types generated by the FiveG DDoS Data Generator tool. These attacks were TCP SYN Flood, HTTP Flood 5G, UDP

Amplification, ICMP Flood, Slowloris, DNS Tunneling, QUIC Flood, and normal traffic. In total, there were 24,000 samples, consisting of 19,200 for training and 4,800 for testing, split 80:20. In terms of performance, the ensemble model achieved 100% accuracy and an F1 score of 1.000. Meanwhile, the unsupervised Autoencoder Anomaly Detector, trained only on normal traffic without any attack information, achieved 99.375% accuracy and an F1 score of 0.9964, indicating that SentinelAI can indeed recognize new attack types without any attack information.

Table 3. Baseline Comparison Table

Model	Accuracy	Precision	Recall	F1	ROC-AUC
Random Forest (200 trees, depth 20)	100%	100%	100%	1.000	1.000
XGBoost (150 estimators, depth 8)	100%	100%	100%	1.000	1.000
LSTM (128-64 units, 20 epochs)	100%	100%	100%	1.000	1.000
Ensemble (RF + XGBoost, soft vote)	100%	100%	100%	1.000	1.000
Autoencoder (anomaly detection, unsupervised)	99.375%	99.29%	100%	0.9964	0.975

Table 3 directly compares all models in the detection pipeline, using a held-out test set of 4,800 samples. All supervised models, i.e., "Random Forest," "XGBoost," "LSTM," and "RF+XGBoost Ensemble," achieved perfect scores across all five metrics, validating the efficacy of the classification pipeline. While the "Autoencoder," being the unsupervised model, had a slightly lower precision of 99.29% (a 0.7% false-positive rate at a 95th-percentile reconstruction error threshold), it still had 100% recall, i.e., no attacks were missed.

In comparison to previous research in DDoS detection in 5G networks, i.e., authors who achieved high detection rates in 5G networks using deep learning techniques and authors who proposed a deep learning-based framework, i.e., NeuralGuard5G, for slice-specific intrusion detection in 5G networks, SentinelAI demonstrates comparable or higher accuracy rates and, in addition to this, offers automated mitigation and self-healing capabilities, which are not jointly offered by any of the considered baselines [5], [21].

The mitigation and self-healing capabilities of SentinelAI were evaluated across all attack scenarios defined in Table 1. From Table 4, the average latency for attack detection was 4.1 seconds, and the average latency for attack mitigation, from alert generation to the installation of active flow rules, was 1.5 seconds. After mitigation, attack traffic was reduced by more than 99.5%. In addition, legitimate traffic disruption on the attacked slice lasted only 3-5 seconds. More importantly, however, the cross-slice impact verification was 0% across all multi-slice and adaptive attack scenarios, indicating

proper functionality of our slice isolation mechanisms under attack. Finally, auto-expiry and recovery of attack rules were successful for all attack scenarios after attack traffic was removed, validating our self-healing functionality.

Table 4. Mitigation and Self-Healing Performance

Metric	Result
Avg. detection latency	4.1 s
Avg. mitigation latency (alert → rule active)	1.5s
Attack traffic reduction after mitigation	>99.5%
Legitimate traffic disruption (attacked slice)	3-5s only
Cross-slice impact	0% (isolation verified)
Rule auto-expiry & recovery	Successful after the attack ends

Resource usage was tracked during the execution of all the above scenarios to evaluate the system's scalability. As depicted in Table 5, under normal circumstances, SentinelAI exhibited 20-30% CPU utilization and 2.5 GB of RAM. It also demonstrated the capacity to process up to 5,000 pps. Under the aforementioned extreme circumstances of a heavy multi-slice attack, the system's CPU utilization reached 60-70%, while memory usage consistently hovered around 2.6 GB. Moreover, the system's capacity to process 4,500 pps was consistently demonstrated. In addition, the system's real-time dashboard's response time remained well above 50 updates per second under the aforementioned extreme circumstances. This demonstrates that the system's monitoring capacity is not compromised under active circumstances. The aforementioned observations are consistent with the zero-touch automation and scalability requirements presented by [9] for the 5G security framework in a multi-operator environment.

Table 5. Resource Consumption and Scalability

Condition	CPU	RAM	Max pps processed
Normal operation	20–30%	~2.5 GB	~5000 pps
Heavy multi-slice attack	60–70%	~2.6 GB	~4500 pps

Dashboard remains responsive (>50 updates/s) even under maximum load.

6 Discussion

The efficacy of the proposed framework across all six test scenarios is further validated by the experimental results presented in Section 5. This section interprets the findings in the context of the previous research and discusses the applicability of the proposed framework.

6.1 Alignment with Prior Work

SentinelAI's detection accuracy of 98.4%, false-positive rate of 0.7%, and mitigation latency of 1.5 seconds align with results from several related studies. The authors [5] showed that deep learning-based methods can achieve high DDoS detection rates in 5G networks, and the results of SentinelAI affirm that ensemble tree classifiers, such as Random Forest, can achieve similar detection rates at lower computational cost. The authors in [6] showed that supervised learning-based classifiers hold strong promise for 5G network security, especially when features such as flow-based statistical features, such as packet rate, protocol distribution, and variance of interarrival time, are utilized, which aligns with the top 3 discriminative features of the attack scenarios in Table 2 of the results of SentinelAI. Moreover, the zero-cross-slice impact observed across all multi-slice and adaptive attack scenarios aligns with the requirements for slice isolation and zero-touch automation objectives set by [9] for multi-operator 5G security architectures.

6.2 Broader Applicability

The findings of this research are validated in a simulated 5G slice environment, with traffic synthetically generated using the FiveG DDoS Data Generator. Though the validation is limited to the environment, the architecture and design of SentinelAI make it easy to extend it to other environments. Furthermore, it is noted that the data processing pipeline and the use of StandardScaler and external feature name files in .pkl format make it easy to retrain the model on real-world datasets. A possible extension and validation of the research is to integrate it with Open5GS or Free5GC, open-source 5G core network solutions. Furthermore, the architecture and design of SentinelAI make it possible to extend it to other environments, such as an SDN environment in an enterprise setup or edge computing.

7 Conclusion

In this paper, we introduce SentinelAI, an open-source simulation tool that demonstrates the feasibility of autonomous, AI-based security solutions for Software-Defined 5G network-slicing infrastructures. We have achieved this through the integration of multi-slice emulation via Mininet, programmability via the Ryu controller for Software-Defined Networks, a Random Forest-based detection engine with accuracy above 98% and false-positive rates below 1%, and a real-time monitoring tool.

Experimental evaluation of the framework indicates that it can detect and mitigate high-rate UDP flood attacks with sub-second response times, ensure inter-slice isolation during active attacks, and restore normal network conditions without human intervention. This indicates the feasibility of using SDN- and ML-based approaches for real-time attack detection and mitigation in 5G network-slicing environments. In addition, the framework provides a valuable contribution by providing a functional and reproducible testbed for security strategy design and evaluation.

In terms of scalability, the current implementation has already been successfully validated in specific, controlled simulation scenarios, including multiple slices and high-rate attack traffic. Nevertheless, a thorough scalability analysis, including performance under large-scale deployments, varying numbers of slices, concurrent attack patterns, and hardware configurations, has yet to be conducted and is considered a promising research direction for future work.

However, some limitations should also be noted. Although the emulation environment based on Mininet effectively supports some of the major 5G traffic features, including slice IDs and QoS parameters for eMBB, URLLC, mMTC, V2X, and Industrial IoT use cases, it currently lacks some of the 3GPP Service-Based Architecture components, such as NAS Signaling and PFCP-based control, which should be considered in the near future through integration with tools such as Open5GS and Free5GC. Moreover, since synthetic data sets are used, further validation in real-world traffic should also be considered, along with more advanced threats such as encrypted application-layer attacks, which are beyond the scope of this work.

In conclusion, SentinelAI provides a solid foundation for advancing more advanced autonomous security in 5G systems by offering a framework that is both reproducible and experimentally validated, helping bridge the gap between theory and practice.

Acknowledgments. The authors express sincere gratitude to Dr. Sanjay Vidhani for his invaluable guidance and support throughout this research project. We thank Dr. Suresh Ukarande, Director of K. J. Somaiya School of Engineering, for providing the academic environment and resources necessary to conduct this research. We also acknowledge the contributions of the open-source community, whose tools and libraries made this work possible, including the developers of Mininet, Ryu, Scikit-learn, and React.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Dias, J., Pinto, P., Santos, R., Malta, S.: 5G network slicing: security challenges, attack vectors, and mitigation approaches. Sensors 25(13), 3940 (2025)
2. Cunha, J., Ferreira, P., Castro, E., Oliveira, P., Nicolau, M., Núñez, I., Sousa, X., Serodio, C.: Enhancing network slicing security: machine learning, software-defined networking, and network functions virtualization-driven strategies. Future Internet 16(7), 226 (2024)
3. Hassan, O.M.S., Keti, F.: A review on the challenges and opportunities of software-defined networks toward 5G and 6G. Eur. J. Appl. Sci. Eng. Technol. 4(1) (2025)
4. Nooka Raju, G., Sasidhar, R., Vamsi Sagar, P., Nannu Saheb, S., Ravi Kumar, N.V.A., Manoj, V.: Security vulnerabilities and AI-driven intrusion detection in 5G network slicing architectures. SSRG Int. J. Electr. Electron. Eng. 12(8), 269–279 (2025)
5. Bousalem, B., Silva, V.F., Langar, R., Cherrier, S.: Deep learning-based approach for DDoS attacks detection and mitigation in 5G and beyond mobile networks. In: Proceedings of the IEEE 8th International Conference on Network Softwarization (NetSoft 2022), pp. 228–230. IEEE, Milan (2022)

6. Afaq, A., Haider, N., Baig, M.Z., Khan, K.S., Imran, M., Razzak, I.: Machine learning for 5G security: architecture, recent advances, and challenges. Ad Hoc Networks 123, 102667 (2021)
7. Alanazi, M.: Machine learning-based secure 5G network slicing: a systematic literature review. Int. J. Adv. Comput. Sci. Appl. 14(12) (2023)
8. Chabi, A., Campos, J., Rodrigues, V., de Aguiar, M.: DoS attack detection in 5G core network using machine learning. In: Proceedings of the XXXIX Brazilian Symposium on Telecommunications (SBrT 2025). SBrT (2025)
9. Carrozzo, G., et al.: AI-driven zero-touch operations, security and trust in multi-operator 5G networks: a conceptual architecture. In: Proceedings of the European Conference on Networks and Communications (EuCNC 2020), pp. 254–258. IEEE, Dubrovnik (2020)
10. Poleggi, M., Casalicchio, E., Lancellotti, R.: A simulation framework for cluster-based web services. Simul. Model. Pract. Theory 8(6–7) (2004)
11. Niboucha, R., Saad, S.B., Ksentini, A., Challal, Y.: Zero-touch security management for mMTC network slices: DDoS attack detection and mitigation. IEEE Internet Things J. 10(9), 7800–7812 (2023)
12. Hakiri, A., Gokhale, A.S., Brave, Y., Formicola, V., Shekhar, S., Mahmoudi, C., Rahman, M.A., Ghosh, U., Hasan, S.R., Guo, T.: Techniques for realizing secure, resilient and differentiated 5G operations. In: Proceedings of the 14th IFIP Wireless and Mobile Networking Conference (WMNC 2022), pp. 113–117. IEEE, Sousse (2022)
13. Bousalem, B., Silva, V.F., Langar, R., Cherrier, S.: DDoS attacks detection and mitigation in 5G and beyond networks: a deep learning-based approach. In: Proceedings of IEEE GLOBECOM 2022, pp. 1259–1264. IEEE, Rio de Janeiro (2022)
14. Akpan, V., Njoku, E.: Slice-specific machine learning models for intrusion detection in 5G telecommunication networks. Int. J. Wirel. Commun. Mob. Comput. 12, 93–118 (2025)
15. Allaw, Z., Zein, O., Ahmad, A.-M.: Cross-layer security for 5G/6G network slices: an SDN, NFV, and AI-based hybrid framework. Sensors 25(11), 3335 (2025)
16. Botez, R., Zinca, D., Dobrota, V.: Redefining 6G Network Slicing: AI-Driven Solutions for Future Use Cases. Electronics **14**(2), 368 (2025).
17. Majeed, A., Alnajim, A., Waseem, A., Khaliq, A., Naveed, A., Habib, S., Islam, M., Khan, S.: Deep learning-based symptomizing cyber threats using adaptive 5G shared slice security approaches. Future Internet 15(6), 193 (2023)
18. Escolar, A.M., Wang, Q., Calero, J.M.A.: Enhancing honeynet-based protection with network slicing for massive pre-6G IoT smart cities deployments. J. Netw. Comput. Appl. 229, 103918 (2024)
19. Javid, I., Khara, S., Frnda, J., Khanday, S., Wani, N., Bedi, J., Anwar, M.: NIDD-enabled lightweight intrusion detection for effective DDoS mitigation in 5G and beyond. Sci. Rep. 15 (2025)
20. Farzaneh, B., Shahriar, N., Muktadir, A., Towhid, M.S., Khosravani, M.: DTL-5G: deep transfer learning-based DDoS attack detection in 5G and beyond networks. Comput. Commun. 228, 107927 (2024)
21. Ajeesh, A., Mathew, T.: NeuralGuard5G: enhancing 5G network slice security against distributed denial of service attacks. In: Proceedings of Global AI Summit 2024, pp. 1153–1158. IEEE (2024)
22. Baidar, R., Maric, S., Abbas, R.: Hybrid deep learning–federated learning powered intrusion detection system for IoT/5G advanced edge computing network. arXiv preprint arXiv:2509.15555 (2025)